SCIENTIFIC REPORTS
nature research

OPEN

Accelerated spin dynamics using deep learning corrections

Sojeong Park[1,2], Wooseop Kwak[1] & Hwee Kuan Lee[2,3,4,5]✉

Theoretical models capture very precisely the behaviour of magnetic materials at the microscopic level. This makes computer simulations of magnetic materials, such as spin dynamics simulations, accurately mimic experimental results. New approaches to efficient spin dynamics simulations are limited by integration time step barrier to solving the equations-of-motions of many-body problems. Using a short time step leads to an accurate but inefficient simulation regime whereas using a large time step leads to accumulation of numerical errors that render the whole simulation useless. In this paper, we use a Deep Learning method to compute the numerical errors of each large time step and use these computed errors to make corrections to achieve higher accuracy in our spin dynamics. We validate our method on the 3D Ferromagnetic Heisenberg cubic lattice over a range of temperatures. Here we show that the Deep Learning method can accelerate the simulation speed by 10 times while maintaining simulation accuracy and overcome the limitations of requiring small time steps in spin dynamic simulations.

Magnetic materials have a wide range of industrial applications such as in Nd–Fe–B-type permanent magnets used for motors in hybrid cars[1,2], magnetoresistive random access memory (MRAM) based on the storage of data in stable magnetic states[3], ultrafast spins dynamics in magnetic nanostructures[4,5], heat assisted magnetic recording and ferromagnetic resonance methods for increasing the storage density of hard disk drives[6,7], exchange bias related to magnetic recording[8], and magnetocaloric materials for refrigeration technologies[1]. Understanding the underlying physics of magnetic material enables us to develop much better applications. In particular, the study of the properties of these magnetic materials is performed experimentally by using neutron scattering[9]. Magnetic properties of materials are also studied theoretically using computational methods. Spin dynamics simulations[10] are powerful tools for understanding fundamental properties of magnetic materials that can be verified by experimental methods. In spin dynamics simulations, classical equations of motion of spin systems are solved numerically using well known integrators such as leapfrog, Verlet, predictor-corrector, and Runge-Kutta methods[11–13]. The accuracy of these simulations depends on a time integration step size. If a large time step is used, the accumulated truncation error becomes larger. Conversely, using a short time step is very computationally demanding. So, it is important to find a trade off between speed and accuracy.

Symplectic methods[14,15] are among the most useful time integrators for spin dynamics simulations. The numerical solutions of symplectic methods have properties of the time reversibility and the energy conservation. For example, high order Suzuki–Trotter decomposition method, one of the symplectic methods, allows for larger time step with limited error in its computation. In this paper, we seek to enhance the time integration step of Suzuki–Trotter decomposition method further using Deep Learning techniques. For second-order Suzuki–Trotter decomposition method, the integration time step is limited up to $\tau \sim 0.04/J$ and for fourth-order Suzuki–Trotter decomposition method, the integration time step is limited up to $\tau \sim 0.2/J$[16].

Recently, Machine Learning techniques are used to enhance simulation efficiencies in the condensed matter physics. Its applications include addressing difficulties of phase transition[17–22] and accelerating the Monte-Carlo simulations[23]. A crucial issue in molecular dynamics simulations[24] is that generating samples from the equilibrium distributions is time consuming. Boltzmann generators machine[25] addresses the long-standing rare-event (e.g. transition) sampling problem. In addition, study of quantum many body systems using Machine Learning

[1]Department of Physics, Chosun University, Gwangju 61452, Republic of Korea. [2]Bioinformatics Institute, Agency for Science, Technology and Research (A*STAR), 30 Biopolis Street, #07-01 Matrix, Singapore 138671, Singapore. [3]School of Computing, National University of Singapore, 13 Computing Drive, Singapore 117417, Singapore. [4]Singapore Eye Research Institute (SERI), 11 Third Hospital Ave, Singapore 168751, Singapore. [5]Image and Pervasive Access Laboratory (IPAL), 1 Fusionopolis Way, #21-01 Connexis (South Tower), Singapore 138632, Singapore. ✉email: leehk@bii.a-star.edu.sg

is applied to simulation of the quantum spin dynamics[26,27], identifying phase transitions[28], and solves the exponential complexity of the many body problem in quantum systems[29].

In this paper, we show that speed up is achieved if we combine spin dynamics simulation and Deep Learning to learn the error corrections. The first condition for speed up is enough capacity of Deep Learning to learn the associations between spin configuration generated by large time steps and spin configuration generated by accurate short time steps. The second condition is enough training data for learning and show the Deep Learning enough pairs of patterns between spin configuration for large and short time steps. We propose to use Deep Learning to estimate the error correction terms of Suzuki–Trotter decomposition method, and then add the correction terms back to spin dynamics results, making them more accurate. As a result of this correction, larger time step can be used for Suzuki–Trotter decomposition method, and corrections can be made for each time step. To evaluate our Deep Learning method, we analyze spin-spin correlation as a more stringent measure. We also use thermal averages to benchmark the performance of our method. We compare the Deep Learning results with those from spin dynamics simulation without Deep Learning for short time steps.

Methods

Heisenberg model. The ferromagnetic Heisenberg model on a cubic lattice is used to demonstrate the efficiency of our method. The Hamiltonian for this model is given as $H = -J\sum_{<i,j>} \boldsymbol{S}^i \cdot \boldsymbol{S}^j$, where a vector $\boldsymbol{S}^i$ has three components (S^i_x, S^i_y, S^i_z) and $|\boldsymbol{S}^i|$ is a unit vector. We formalize our spin dynamics following the notations of Tsai *et al.*[16]. We write the equations of motion for all spins as

$$\frac{d\sigma(t)}{dt} = \hat{R}\sigma(t), \tag{1}$$

where $\sigma(t) = (\boldsymbol{S}^1(t), \boldsymbol{S}^2(t), \ldots, \boldsymbol{S}^n(t))$ is the spin configuration at time t. The integration of the equations of motion in Eq. (1) is done using the second order Suzuki–Trotter decomposition method as in Tsai *et al.*[16]. As following the mathematical notations of Tsai et al., we decompose the evolution operator $\hat{R}$ into $\hat{R}_A$ and $\hat{R}_B$ on the sublattices A and B respectively, and obtain

$$e^{(\hat{R}_A+\hat{R}_B)\tau} = e^{\hat{R}_B\tau/2}e^{\hat{R}_A\tau}e^{\hat{R}_B\tau/2} + O(\tau^3) \tag{2}$$

The ferromagnetic Heisenberg model is considered on the cubic lattice of dimensions $L \times L \times L$ with periodic boundary conditions. This model undergoes a phase transition at a temperature $k_BT_c/J = 1.442\ldots$[30], where k_B is Boltzmann's constant. In the spin dynamics approach, the equations of motion for the Heisenberg model is governed by the following equation:

$$\frac{d\boldsymbol{S}^i}{dt} = -\boldsymbol{S}^i \times \boldsymbol{H}^i_{\text{eff}} = \begin{bmatrix} 0 & -H^i_{\text{eff,z}} & H^i_{\text{eff,y}} \\ H^i_{\text{eff,z}} & 0 & -H^i_{\text{eff,x}} \\ -H^i_{\text{eff,y}} & H^i_{\text{eff,x}} & 0 \end{bmatrix} \boldsymbol{S}^i = R^i\boldsymbol{S}^i. \tag{3}$$

Here, $\boldsymbol{H}^i_{\text{eff}}$ is the effective field acting on the ith spin. The k component of the effective field can be specified as $\boldsymbol{H}^i_{\text{eff,k}} = -\sum_{j=nn(i)} S^j_k$, where the sum runs over the nearest neighbor pairs of sites and $k = x, y$, and z.

Deep Learning approach. A fully supervised Deep Learning method is developed to perform the spin dynamics by using the second order Suzuki–Trotter decomposition method to reduce simulation errors. In order to produce training data for our supervised Deep Learning, initial spin configurations are considered at ordered, near-critical, and disordered states in the temperature range $k_BT/J \in [0.5, 2.4]$ and sampling 9.1×10^5 independent spin configurations using Monte-Carlo simulations with the Metropolis–Hastings algorithm[30–33]. The initial spin configurations are prepared with 300,000 samples in ordered states, 210,000 samples near critical states, and 400,000 samples disordered states by simulated annealing method. The temperature annealing scheme will be described in more details in the supplementary information. The temperatures for annealing are gradually lowered from high to low temperatures and Monte Carlo data are always obtained at equilibrium configurations. For each sampled initial spin configuration σ_i, two sets of spin dynamics simulations are performed with the time steps $\tau_1 = 10^{-1}$ and $\tau_3 = 10^{-3}$ as illustrated in Fig. 1a. Second-order Suzuki–Trotter method uses $\tau = 0.04$ as typical integration time step, so we use $\tau = 10^{-3}$ which would give good accurate simulation. For large time step, we tried $\tau = 10^{-2}$ and $\tau = 10^{-1}$, with our Deep Learning corrections, a large time step of $\tau = 10^{-1}$ gives the best speed up with a good accuracy. The spin configuration with time step $\tau_3 = 10^{-3}$ needs 100 time steps of simulations to pair with the spin configuration with one time step $\tau_1 = 10^{-1}$. Formally, we represent the updated spin configurations $\sigma_i^{(10^{-1})}$ and $\sigma_i^{(10^{-3})}$ by using the Suzuki–Trotter decomposition method as

$$\begin{aligned} \sigma_i^{(10^{-1})} &\leftarrow e^{\hat{R}_B\tau_1/2}e^{\hat{R}_A\tau_1}e^{\hat{R}_B\tau_1/2}\sigma_i, \qquad \tau_1 = 10^{-1} \\ \sigma_i^{(10^{-3})} &\leftarrow (e^{\hat{R}_B\tau_3/2}e^{\hat{R}_A\tau_3}e^{\hat{R}_B\tau_3/2})^{100}\sigma_i, \ \tau_3 = 10^{-3} \quad i = 1, \ldots D, \end{aligned} \tag{4}$$

where σ_i is an initial spin configuration and D represents the number of training data. The difference between spin configuration $\sigma_i^{(10^{-3})}$ generated using $\tau_3 = 10^{-3}$ and spin configuration $\sigma_i^{(10^{-1})}$ generated using $\tau_1 = 10^{-1}$ is captured by

$$\sigma_i^{(res)} = \sigma_i^{(10^{-3})} - \sigma_i^{(10^{-1})} \qquad i = 1, \ldots, D\,, \tag{5}$$

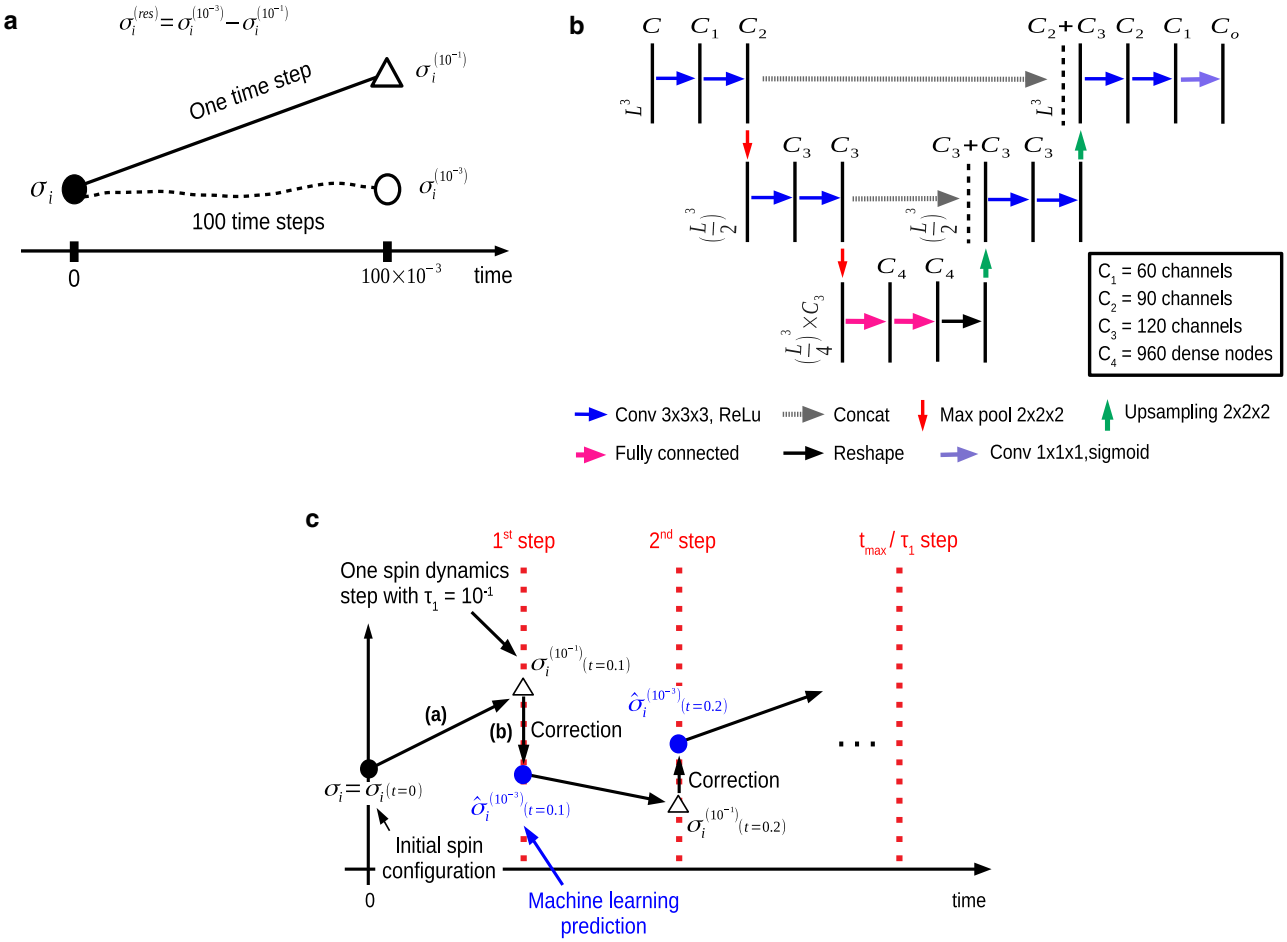

Figure 1. Deep learning for Heisenberg model. **a** Spin configurations for training data preparation. σ_i is initial spin configuration, $\sigma_i^{(10^{-1})}$ is spin configuration after one time step of $\tau_1 = 10^{-1}$ from σ_i, and $\sigma_i^{(10^{-3})}$ is spin configuration after 100 time steps of $\tau_3 = 10^{-3}$ from σ_i. $\sigma_i^{(res)}$ is residue of $\sigma_i^{(10^{-3})}$ and $\sigma_i^{(10^{-1})}$. **b** Illustration of the U-Net architecture. Each vertical black line represents a multi-channel feature map. The number of channels is denoted on the top of the straight vertical black line and each map's dimension is indicated on the left edge. Vertical dashed black lines correspond on the copied feature maps from each encoder layer. **c**, A sequence of spin dynamics for testing the trained U-Net model: (a) conduct one time step $\tau_1 = 10^{-1}$ of spin dynamics simulation; (b) use $\sigma_i^{(10^{-1})}$ to predict the spin configuration $\sigma_i^{(10^{-3})}$ by estimating predicted residue $\hat{\sigma}_i^{(res)}$ using Eq. (6); Steps (a) and (b) are repeated up to t_{max} time.

where $\sigma_i^{(res)}$ is residue. For our Deep Learning, initial spin configuration σ_i and spin configuration $\sigma_i^{(10^{-1})}$ are used as the inputs into U-Net[34], a kind of convolutional neural networks. The U-Net is a proven architecture for image segmentation as well as for extracting subtle features. The detailed structure of U-Net is shown in Fig. 1b. The architecture of U-net used for $8 \times 8 \times 8$ cubic lattice is that convolutional layers are used as an encoder on left upper side followed by a decoder on right upper side that consists of upsamplings and concatenations with the correspondingly feature maps from the encoder. We add fully connected layers (FC) in the bottom of the network between the encoder and the decoder to efficiently determine particular weights in the feature map from the encoder , such as capturing more information of spin-spin interactions. The input channels C are 6 by concatenating spin coordinates S_x, S_y, and S_z of both σ_i and $\sigma_i^{(10^{-1})}$, respectively. The input dimensions of U-Net are reshaped to $[D, L, L, L, C]$ as cubic grid vector map, where D is the total number of training data, L is lattice size, and C is input channels. The encoder consists of the repeated two convolutional layers with $3 \times 3 \times 3$ filters followed by a $2 \times 2 \times 2$ max pooling. We apply a reshaping function to FC with dimensions from $[D, \frac{L}{4} \times \frac{L}{4} \times \frac{L}{4} \times C_4]$ into $[D, \frac{L}{4}, \frac{L}{4}, \frac{L}{4}, C_4]$. Every step in decoder consists of upsampling layers with a $2 \times 2 \times 2$ filters followed by the repeated two convolutional layers with $3 \times 3 \times 3$ filters and copies with correspondingly cropped feature maps from encoding layers. The periodic boundary conditions are also applied to the convolutional layers. The activation function of the output is a sigmoid for predicting values of residue with $[D, L, L, L, C_o]$ dimensions, where the number of output channels C_o is 3. A simpler U-Net architecture is used for $4 \times 4 \times 4$ cubic lattice (see supplementary information).

Deployment of our U-Net for spin dynamics. To deploy the trained U-Net for spin dynamics, spin dynamics simulation is carried out with one large time step $\tau_1 = 10^{-1}$ and this simulation result $\sigma_i^{(10^{-1})}$ can be used to predict $\sigma_i^{(10^{-3})}$ as follows:

$$\hat{\sigma}_i^{(10^{-3})} = \sigma_i^{(10^{-1})} + \hat{\sigma}_i^{(res)} \simeq \sigma_i^{(10^{-3})}, \tag{6}$$

where $\hat{\sigma}_i^{(10^{-3})}$ is the predicted spin configuration for 100 time steps of $\tau_3 = 10^{-3}$ and predicted residue $\hat{\sigma}_i^{(res)}$ is the correction term by Deep Learning. A sequence of spin dynamics are conducted at $\tau_1 = 10^{-1}$ and for each step, Eq. (6) is used to perform corrections as shown in Fig. 1c. This new time integration scheme is repeated up to maximum time t_{max}. This scheme requires only forward propagation using the GPU implemented with TensorFlow library[35], so the computing time is negligible.

Normalization of residue. The difference between spin configuration generated with $\tau_3 = 10^{-3}$ and that generated with $\tau_1 = 10^{-1}$ is captured by residue $\sigma_i^{(res)}$ in Eq. (5). Let $(\sigma_i^{(res)})_k^j$ be the k component of residual spin at site j of the lattice, and k denotes x, y, and z components. The values of $(\sigma_i^{(res)})_k^j$ can be quite small for some simulations, to maintain numerical stability, we normalize these values as follows. Each component $(\sigma_i^{(res)})_k^j$ over D samples of training data is normalized to a range of [0,1] by fitting to have a Gaussian distribution, and find the mean and standard deviation for each k component, respectively.

For lattice size $L = 4$, $\lambda_{min} = -0.22455$ and $\lambda_{max} = 0.22455$ are defined by taking 11 times the largest standard deviation of k component. 11 standard deviations translates to a p-value of 1.911×10^{-28}, which ensures that during inference, the normalized residue $(\sigma_i^{(res)})_k^j$ is always within the range [0,1]. For lattice size $L = 8$, $\lambda_{min} = -0.25472$ and $\lambda_{max} = 0.25472$ are defined by taking 13 times the largest standard deviation of k component. Finally, each component $(\sigma_i^{(res)})_k^j$ is normalized to the range [0, 1] and guarantee stable convergence of weights and biases in Deep Learning as follows :

$$(\sigma_i^{norm})_k^j = \frac{(\sigma_i^{(res)})_k^j - \lambda_{min}}{\lambda_{max} - \lambda_{min}} \quad (k = x, y, z,\ i = 1, ...D). \tag{7}$$

During the prediction, $(\sigma_i^{(res)})_k^j$ from test data is normalized to a range of [0, 1] by using λ_{min} and λ_{max}, which have already been obtained.

Loss function and training. The loss function for one data point of $(\sigma_i, \sigma_i^{(10^{-1})}, \sigma_i^{norm})$ is the mean-square error between the normalized residue σ_i^{norm} and the predicted normalized residue $\hat{\sigma}_i^{norm}$ and is defined as

$$\mathcal{L}(\sigma_i, \sigma_i^{(10^{-1})}, \sigma_i^{norm}) = \frac{1}{L^3} \sum_{j=1}^{L^3} \left\| (\sigma_i^{norm})^j - (\hat{\sigma}_i^{norm})^j \right\|_2^2, \tag{8}$$

where j is the index of lattice sites. The distance function between the j^{th} site of σ_i^{norm} and the j^{th} site of $\hat{\sigma}_i^{norm}$ is the sum of the square difference of all spin components :

$$\left\| (\sigma_i^{norm})^j - (\hat{\sigma}_i^{norm})^j \right\|_2^2 = \sum_{k=x,y,z} \left((\sigma_i^{norm})_k^j - (\hat{\sigma}_i^{norm})_k^j \right)^2, \tag{9}$$

where i is the index of training data.

Converting $\hat{\sigma}_i^{norm}$ to $\hat{\sigma}_i^{(res)}$. For our Deep Learning, inputs into U-Net are obtained initial spin configurations σ_i and spin configurations $\sigma_i^{(10^{-1})}$ generated by spin dynamics simulations, and output is $\hat{\sigma}_i^{norm}$. We finally predict the spin configuration for 100 time steps of $\tau_3 = 10^{-3}$ using trained Deep Learning model as $\hat{\sigma}_i^{(10^{-3})} = \sigma_i^{(10^{-1})} + \hat{\sigma}_i^{(res)}$, where the predicted residue $\hat{\sigma}_i^{(res)}$ can be obtained by the following converting formula as $\hat{\sigma}_i^{(res)} = \hat{\sigma}_i^{norm}(\lambda_{max} - \lambda_{min}) + \lambda_{min}$.

Results

The effectiveness of our proposed Deep Learning method is evaluated at $k_BT/J = 0.4 < k_BT_c/J$, $k_BT/J = 1.44 \approx k_BT_c/J$, and $k_BT/J = 2.4 > k_BT_c/J$. Note that at $k_BT/J = 2.4$, the system is in a disordered state and spatial corrections between spins are very short. One hundred independent spin configurations are generated by using Monte-Carlo simulation for use as test data sets at each temperature $k_BT/J = 0.4$, 1.44, and 2.4. Second order Suzuki–Trotter decomposition methods are used for all experiments in this paper.

To evaluate the accuracy of simulation results, correlation is investigated by comparing spin dynamics trajectory $\sigma(t)$ with highly accurate spin dynamics trajectory $\rho(t)$ performed with $\tau = 10^{-6}$. $\tau = 10^{-6}$ is used as the reference time step as we found that it can give accurate trajectories. Correlation $\xi(t)$ as function of time t in which $\sigma(t)$ and $\rho(t)$ are compared is given by

$$\xi(\sigma, t) = \frac{1}{L^3} \sum_{j=1}^{L^3} [(\rho^j(t))_x (\sigma^j(t))_x + (\rho^j(t))_y (\sigma^j(t))_y + (\rho^j(t))_z (\sigma^j(t))_z], \tag{10}$$

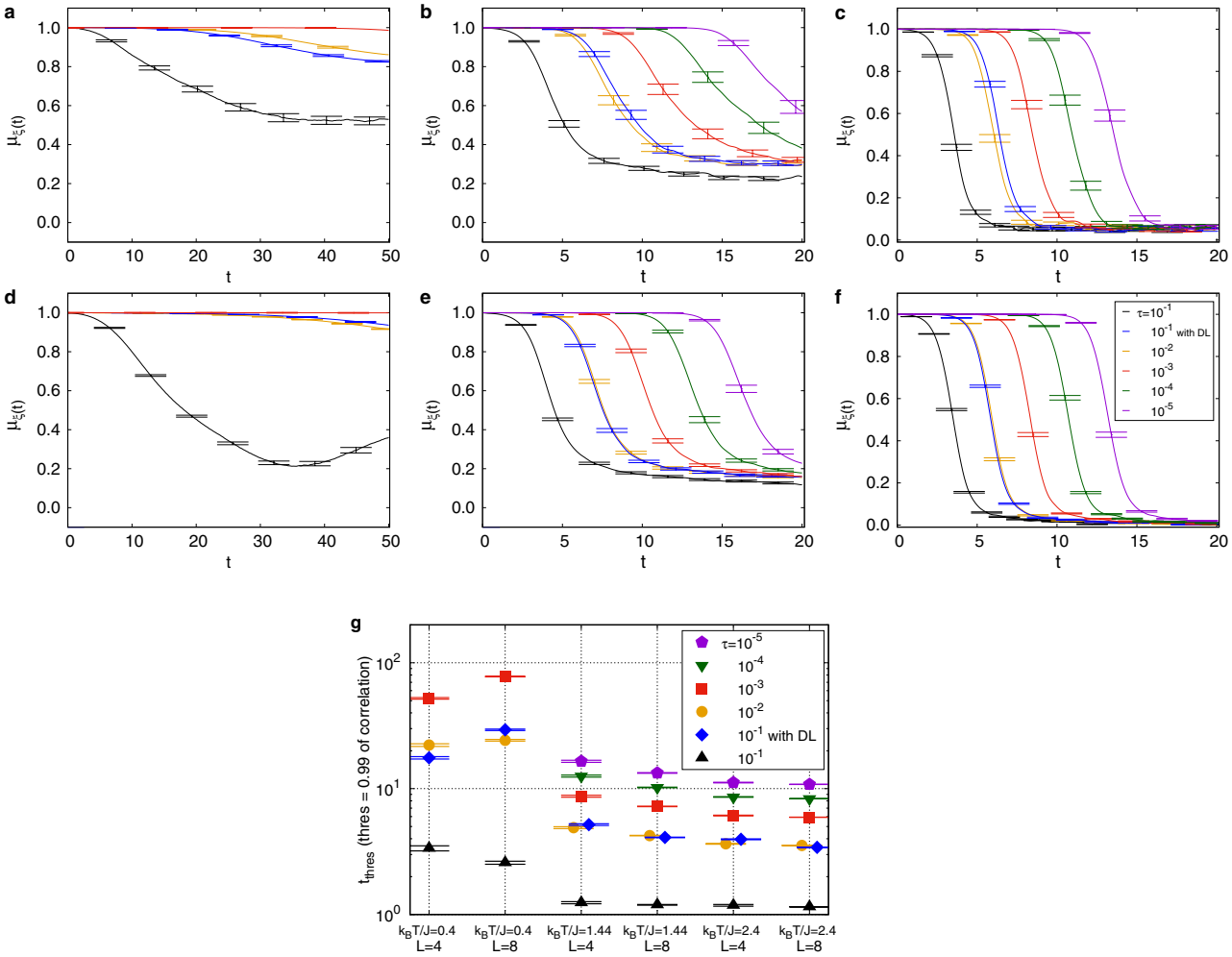

Figure 2. Spin-spin correlation using reference trajectory generated at $\tau = 10^{-6}$. Analysis of the mean of correlation $\mu_{\xi(t)}$ as a function of time on $4 \times 4 \times 4$ cubic lattice at **a**, $k_BT/J = 0.4$, **b**, $k_BT_c/J \approx 1.44$, and **c**, $k_BT/J = 2.4$ and those on $8 \times 8 \times 8$ cubic lattice at **d**, $k_BT/J = 0.4$, **e**, $k_BT_c/J \approx 1.44$, and **f**, $k_BT/J = 2.4$. Blue line presents the Deep Learning (DL) result while black line, yellow line, and red line are the simulation results for $\tau = 10^{-1}$, $\tau = 10^{-2}$, and $\tau = 10^{-3}$, respectively. Especially, at $k_BT/J = 1.44$ and $k_BT/J = 2.4$, green line and violet line show the simulation results for $\tau = 10^{-4}$ and $\tau = 10^{-5}$, respectively. **g** Threshold time t_{thres} as function of temperature. Filled rhombi (◆) represents the Deep Learning result while filled black triangles(▲), filled yellow circles (●), filled red squares (■), filled green inverted triangles (▼), and filled violet pentagons (⬟) are the simulation results without DL corrections for $\tau = 10^{-1}$, $\tau = 10^{-2}$, $\tau = 10^{-3}$, $\tau = 10^{-4}$, and $\tau = 10^{-5}$, respectively.

where index j denotes lattice site of spins, L is the linear dimension of the lattice, and L^3 is total number of spins at lattice sites. Since the initial spin configurations are the same, $\rho(0)$ is identical to $\sigma(0)$. We compute one hundred correlation $\xi(\sigma_i, t)$ for spin configurations $\sigma_i(t)$, where i is from 1 to 100. Then, we also estimate the mean of correlation $\mu_{\xi(t)}$ and the standard deviation of correlation std$(\xi(t))$ of $\xi(\sigma_i, t)$ as a function of time at each temperature.

Suzuki–Trotter decomposition method provides important properties such as conservation of energy $\boldsymbol{e} = -L^{-3}\sum_{<i,j>}^{L^3} \boldsymbol{S}^i \cdot \boldsymbol{S}^j$ and magnetization $\boldsymbol{m} = L^{-3}\sqrt{\left(\sum_i S_x^i\right)^2 + \left(\sum_i S_y^i\right)^2 + \left(\sum_i S_z^i\right)^2}$, and time reversibility. We wish to compare the conservation of energy and magnetization across one hundred samples, but their starting spin configurations are different. In order to take statistics across the samples, we shift the energy and magnetization of the initial spin configurations to zero. Eq. (11) and Eq. (12) show how we shift the energy per site $e(t)$ and magnetization per site $m(t)$ at each time step t. Here, Q represents the number of samples at each temperature. We use Q as one hundred.

$$\tilde{e}_i(t) = e_i(t) - e_i(0) \quad i = 1, \ldots, Q \tag{11}$$

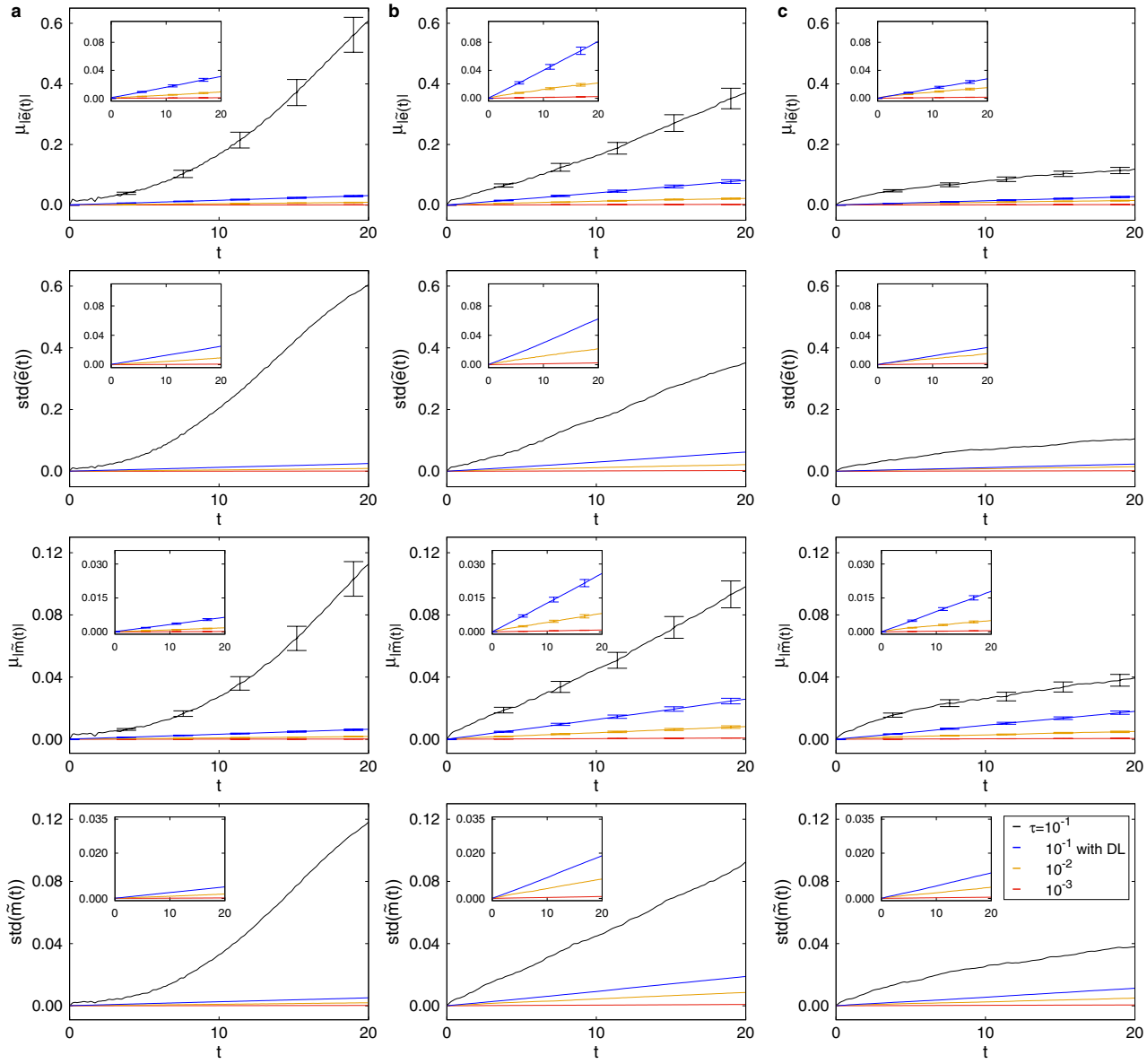

Figure 3. Conservation of energy and magnetization on $4 \times 4 \times 4$ cubic lattice. Predictions of the mean of absolute energy per site $\mu_{|\tilde{e}(t)|}$, standard deviation of energy per site std$(\tilde{e}(t))$, the mean of absolute magnetization per site $\mu_{|\tilde{m}(t)|}$, and standard deviation of magnetization per site std$(\tilde{m}(t))$ as a function of time at **a**, $k_BT/J = 0.4$, **b**, $k_BT_c/J \approx 1.44$, and **c**, $k_BT/J = 2.4$. Black line, yellow line, and red line represent data obtained from spin dynamics simulations with $\tau = 10^{-1}$, $\tau = 10^{-2}$, and $\tau = 10^{-3}$, respectively, while blue line represents data from Deep Learning (DL) correction.

$$\tilde{m}_i(t) = m_i(t) - m_i(0) \quad i = 1, \ldots, Q \tag{12}$$

With the shifting of energy and magnetization, we can compute the mean of absolute energy per site $\mu_{|\tilde{e}(t)|}$, the mean of absolute magnetization per site $\mu_{|\tilde{m}(t)|}$, standard deviation of energy per site std$(\tilde{e}(t))$, and standard deviation of magnetization per site std$(\tilde{m}(t))$ over independent samples.

In Fig. 2, the spin-spin correlation plots are shown as using reference trajectory generated at the reference time step $\tau = 10^{-6}$ for $k_BT/J = 0.4$ ($k_BT/J < k_BT_c/J$) [Fig. 2a,d], $k_BT/J = 1.44$ ($k_BT/J \approx k_BT_c/J$) [Fig. 2b,e], and $k_BT/J = 2.4$ ($k_BT/J > k_BT_c/J$) [Fig. 2c,f]. At $k_BT/J < k_BT_c/J$, correlations remain high (red line, yellow line, and blue line) except for at $\tau = 10^{-1}$ without Deep Learning corrections (black line), where correlation drops around $t = 2$. This is due to accumulation of errors for large time steps. Correlation is recovered with Deep Learning corrections (blue line). Indeed correlations of $\tau = 10^{-1}$ with Deep Learning corrections are as good as for $\tau = 10^{-2}$ without Deep Learning corrections (yellow line), demonstrating a ~ 10 times speed up. At $k_BT/J \approx k_BT_c/J$ and $k_BT/J > k_BT_c/J$, spin-spin correlation drops faster than $k_BT/J < k_BT_c/J$ even for short time steps, $\tau = 10^{-4}$ (green line) and $\tau = 10^{-5}$ (violet line), due to disorder in the spin lattices.

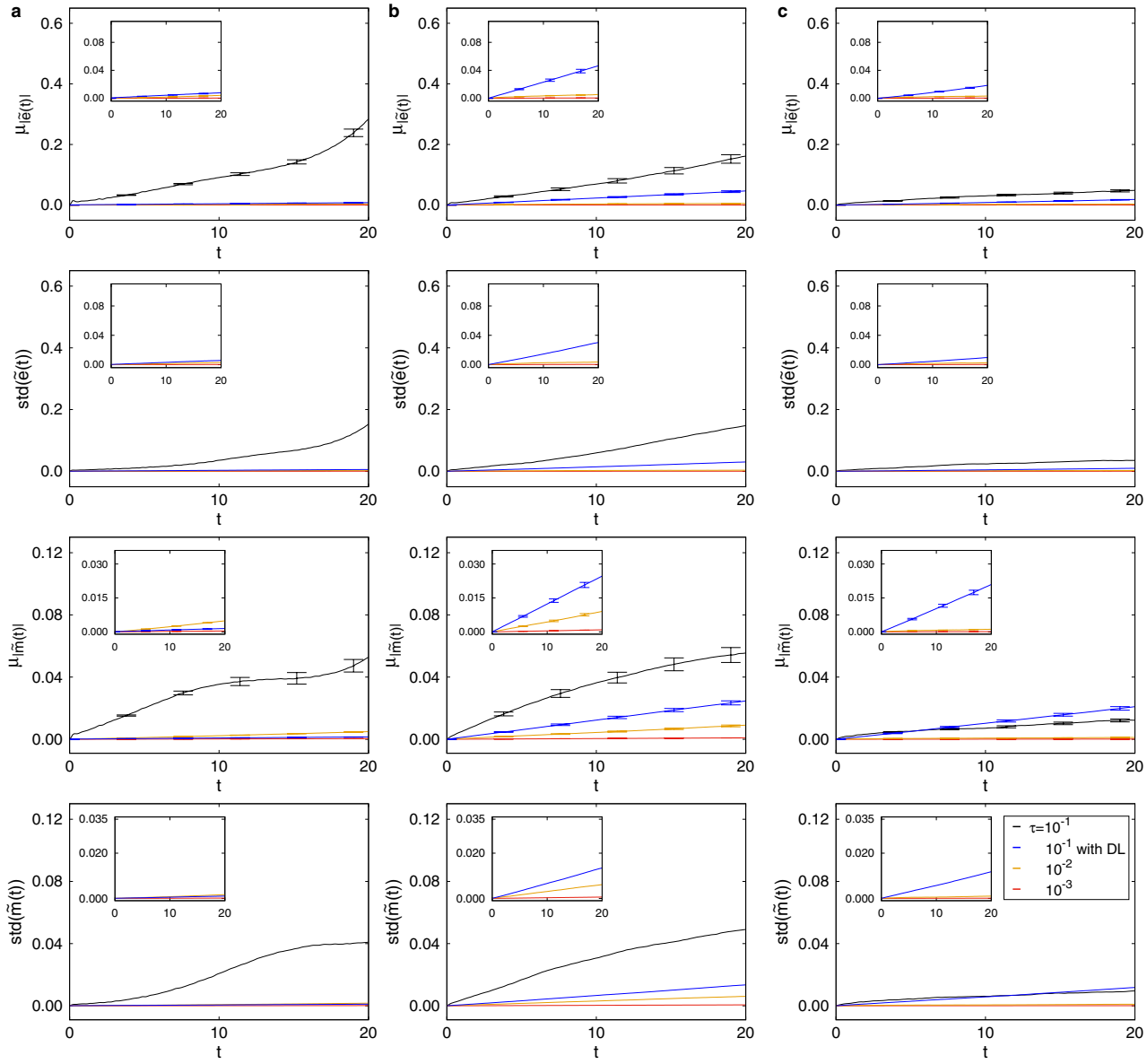

Figure 4. Conservation of energy and magnetization on $8 \times 8 \times 8$ cubic lattice. Predictions of $\mu_{|\tilde{e}(t)|}$, std($\tilde{e}(t)$), $\mu_{|\tilde{m}(t)|}$, and std($\tilde{m}(t)$) as a function of time at **a**, $k_BT/J = 0.4$, **b**, $k_BT_c/J \approx 1.44$, and **c**, $k_BT/J = 2.4$. Black line, yellow line, and red line represent data obtained from spin dynamics simulations with $\tau = 10^{-1}, \tau = 10^{-2}$, and $\tau = 10^{-3}$, respectively, while blue line represents data from Deep Learning (DL) correction. These figures show that the effect of averaging over disordered spins for $L = 8$ is stronger than for $L = 4$ above the critical temperature k_BT_c/J.

We define threshold time t_{thres} as the average time required for spin-spin correlation $\mu_{\xi(t)}$ to drop from 1 to 0.99. In Fig. 2g, the plot of t_{thres} as a function of temperature k_BT/J has the logarithmic scale on the y-axis, and simulations for $\tau = 10^{-3}$ have higher threshold time (red squares) at each temperature than for $\tau = 10^{-1}$ without Deep Learning corrections. Threshold time (filled blue diamonds) for $\tau = 10^{-1}$ with Deep Learning corrections approaches to almost the same threshold time (yellow circles) for $\tau = 10^{-2}$ without Deep Learning corrections at each temperature.

Figure 3 ($L = 4$) and Fig. 4 ($L = 8$) show $\mu_{|\tilde{e}(t)|}$, std($\tilde{e}(t)$), $\mu_{|\tilde{m}(t)|}$, and std($\tilde{m}(t)$) as a function of time at $k_BT/J = 0.4$ ($k_BT/J < k_BT_c/J$) [Figs. 3a and 4a], $k_BT/J = 1.44$ ($k_BT/J \approx k_BT_c/J$) [Figs. 3b and 4b], and $k_BT/J = 2.4$ ($k_BT/J > k_BT_c/J$) [Figs. 3c and 4c]. For time steps $\tau = 10^{-2}$ (yellow line) and $\tau = 10^{-3}$ (red line), conservation of both energy and magnetization is good, as shown by the relatively constant mean plots ($\mu_{|\tilde{e}(t)|}$ and $\mu_{|\tilde{m}(t)|}$) and small standard deviations (std($\tilde{e}(t)$) and std($\tilde{m}(t)$)) across independent simulations. At $k_BT/J < k_BT_c/J$ and $k_BT/J \approx k_BT_c/J$, both energy and magnetization are not conserved in simulations without Deep Learning corrections for time step $\tau = 10^{-1}$ (black line). On the other hand, conservation is recovered

using Deep Learning corrections (blue line). In Fig. 3c, at $k_BT/J > k_BT_c/J$, the system is disordered and the mean of absolute energy $\mu_{|\tilde{e}(t)|}$ and the mean of absolute magnetization $\mu_{|\tilde{m}(t)|}$ become more constant, simply due to averaging of disordered spins. Especially, Fig. 4c shows that at $k_BT/J > k_BT_c/J$, the effect of averaging over disordered spins for $L = 8$ is stronger than for $L = 4$. At high temperature, the number of possible states increase exponentially and hence fitting by Deep Learning corrections is more difficult.

Discussion

Our results have demonstrated that the Deep Learning corrections enhance the time integration step of the original Suzuki–Trotter method and have achieved ~ 10 times computational speed up while maintaining accuracy compared to the original Suzuki–Trotter decomposition method. The nature of local nearest neighbours interactions in the lattice means that convolutional structure of the Deep Neural Network is a nature choice of network architecture. Since convolution is translationally invariant, the effect of lattice size on training our U-Net is not a major concern. For example, between $L = 4$ and $L = 8$ lattices, the time required for training the U-Net parameters increases by about 4 times, which is sub-linear with respect to the number of lattice sites. Our Deep Learning was trained on simulation data at $\tau = 10^{-3}$, however its accuracy performance is equivalent to simulation data at $\tau = 10^{-2}$. This shows that our Deep Learning training has not reached its theoretical limit of a perfect prediction. This theoretical limit can be achieved exactly if we train on an infinite amount of data for an infinite capacity. In practise, Deep Learning methods can not be perfect because the amount of data and the capacity of U-Net are finite. The main source of inaccuracies in our Deep Learning method is that U-Net's output does not fit exactly the labeled data generated at $\tau = 10^{-3}$ and that even if U-Net is able to fit the data it has through training, it may not predict perfectly on the data it has never seen in training. For future work, we will explore the effects of Deep Learning corrections on higher order Suzuki–Trotter decomposition. We will also apply Deep Learning corrections such as off lattice systems and integrators such as velocity-verlet.

Data availability

The data and code that support the findings of this study are available from corresponding authors upon request.

Received: 18 May 2020; Accepted: 16 July 2020
Published online: 13 August 2020

References

1. Gutfleisch, O. *et al.* Magnetic Materials and Devices for the 21st Century: Stronger, Lighter, and More Energy Efficient. *Adv. Mater.* **23**, 821–842. https://doi.org/10.1002/adma.201002180 (2011).
2. Sugimoto, S. Current status and recent topics of rare-earth permanent magnets. *J. Phys. D Appl. Phys.* **44**, 064001. https://doi.org/10.1088/0022-3727/44/6/064001 (2011).
3. Slaughter, J. Materials for Magnetoresistive Random Access Memory. *Annu. Rev. Mater. Res.* **39**, 277–296. https://doi.org/10.1146/annurev-matsci-082908-145355 (2009).
4. Bigot, J.-Y. & Vomir, M. Ultrafast magnetization dynamics of nanostructures: Ultrafast magnetization dynamics of nanostructures. *Ann. Phys.* **525**, 2–30 (2013).
5. Walowski, J. & Münzenberg, M. Perspective: Ultrafast magnetism and THz spintronics. *J. Appl. Phys.* **120**, 140901. https://doi.org/10.1063/1.4958846 (2016).
6. Lee, H. K. & Yuan, Z. Studies of the magnetization reversal process driven by an oscillating field. *J. Appl. Phys.* **101**, 033903. https://doi.org/10.1063/1.2426381 (2007).
7. Kryder, M. *et al.* Heat Assisted Magnetic Recording. *Proc. IEEE* **96**, 1810–1835 (2008).
8. Lee, H. K. & Okabe, Y. Exchange bias with interacting random antiferromagnetic grains. *Phys. Rev. B* **73**, 140403. https://doi.org/10.1103/PhysRevB.73.140403 (2006).
9. Lynn, J. W. Temperature dependence of the magnetic excitations in iron. *Phys. Rev. B* **11**, 2624–2637. https://doi.org/10.1103/PhysRevB.11.2624 (1975).
10. Landau, D. P. & Krech, M. Spin dynamics simulations of classical ferro- and antiferromagnetic model systems: comparison with theory and experiment. *J. Phys. Condens. Matter* **11**, R179–R213. https://doi.org/10.1088/0953-8984/11/18/201 (1999).
11. Frenkel, D. & Smit, B. *Understanding Molecular Simulation: from Algorithms to Applications* 2nd edn, Vol. 50 (Elesiver, Amsterdam, 1996).
12. Beeman, D. Some multistep methods for use in molecular dynamics calculations. *J. Comput. Phys.* **20**, 130–139 (1976).
13. Allen, M. P. & Tildesley, D. J. *Computer Simulation of Liquids* (Clarendon Press, Clarendon, 1988).
14. Kim, S. Time step and shadow Hamiltonian in molecular dynamics simulations. *J. Kor. Phys. Soc.* **67**, 418–422 (2015).
15. Engle, R. D., Skeel, R. D. & Drees, M. Monitoring energy drift with shadow Hamiltonians. *J. Comput. Phys.* **206**, 432–452. https://doi.org/10.1016/j.jcp.2004.12.009 (2005).
16. Tsai, S.-H., Lee, H. K. & Landau, D. P. Molecular and spin dynamics simulations using modern integration methods. *Am. J. Phys.* **73**, 615–624. https://doi.org/10.1119/1.1900096 (2005).
17. Carrasquilla, J. & Melko, R. G. Machine learning phases of matter. *Nat. Phys.* **13**, 431–434. https://doi.org/10.1038/nphys4035 (2017).
18. Zhang, W., Liu, J. & Wei, T.-C. Machine learning of phase transitions in the percolation and X Y models. *Phys. Rev. E* **99**, 032142. https://doi.org/10.1103/PhysRevE.99.032142 (2019).
19. Li, Z., Luo, M. & Wan, X. Extracting critical exponents by finite-size scaling with convolutional neural networks. *Phys. Rev. B* **99**, 075418 (2019).
20. van Nieuwenburg, E., Liu, Y.-H. & Huber, S. Learning phase transitions by confusion. *Nat. Phys.* **13**, 435–439. https://doi.org/10.1038/nphys4037 (2017).
21. Morningstar, A. & Melko, R. G. Deep learning the ising model near criticality. *J. Mach. Learn. Res.* **18**, 5975–5991 (2017).
22. Greitemann, J., Liu, K. & Pollet, L. Probing hidden spin order with interpretable machine learning. *Phys. Rev. B* **99**, 060404. https://doi.org/10.1103/PhysRevB.99.060404 (2019).
23. Huang, L. & Wang, L. Accelerated Monte Carlo simulations with restricted Boltzmann machines. *Phys. Rev. B* **95**, 035105. https://doi.org/10.1103/PhysRevB.95.035105 (2017).
24. Rapaport, D. C. *The Art of Molecular Dynamics Simulation* (Cambridge University Press, Cambridge, 2004).
25. Noé, F., Olsson, S., Köhler, J. & Wu, H. Boltzmann generators: sampling equilibrium states of many-body systems with deep learning. *Science* **365**, eaaw1147. https://doi.org/10.1126/science.aaw1147 (2019).

26. Fabiani, G. & Mentink, J. H. Investigating ultrafast quantum magnetism with machine learning. *SciPost Phys* **7**, 4. https://doi.org/10.1016/j.jcp.2004.12.0090 (2019).
27. Weinberg, P. & Bukov, M. QuSpin: a python package for dynamics and exact diagonalisation of quantum many body systems part I: spin chains. *SciPost Phys.* **2**, 003. https://doi.org/10.1016/j.jcp.2004.12.0091 (2017).
28. Kharkov, Y. A., Sotskov, V. E., Karazeev, A. A., Kiktenko, E. O. & Fedorov, A. K. Revealing quantum chaos with machine learning. *Phys. Rev. B* **101**, 064406 (2020).
29. Carleo, G. & Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **355**, 602–606. https://doi.org/10.1126/science.aag2302 (2017).
30. Chen, K., Ferrenberg, A. M. & Landau, D. P. Static critical behavior of three-dimensional classical Heisenberg models: a high-resolution Monte Carlo study. *Phys. Rev. B* **48**, 3249–3256. https://doi.org/10.1016/j.jcp.2004.12.0092 (1993).
31. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092. https://doi.org/10.1016/j.jcp.2004.12.0093 (1953).
32. Binder, K. The Monte Carlo method for the study of phase transitions: a review of some recent progress. *J. Comput. Phys.* **59**, 1–55. https://doi.org/10.1016/j.jcp.2004.12.0094 (1985).
33. Paauw, T., Compagner, A. & Bedeaux, D. Monte-Carlo calculation for the classical F.C.C. Heisenberg ferromagnet. *Physica A* **79**, 1–17. https://doi.org/10.1016/j.jcp.2004.12.0095 (1975).
34. Ronneberger, O., Fischer, P. & Brox, T. U-Net: convolutional Networks for Biomedical Image Segmentation. https://doi.org/10.1016/j.jcp.2004.12.0096 (2015).
35. Abadi, M. *et al.* Tensorflow: Large-scale machine learning on heterogeneous distributed systems. https://doi.org/10.1016/j.jcp.2004.12.0097 (2016).

Acknowledgements

We are grateful to Mustafa Umit Oner for useful suggestions. We would like to thank Kaicheng Liang, Mahsa Paknezhad, Connie Kou, Shier Nee Saw, Liu Wei, and Kenta Shiina for proof reading our paper and giving valuable comments. This work was supported by the Biomedical Research Council of A*STAR (Agency for Science, Technology and Research), Singapore, and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2020R1A2C1003743). S.J.P. is grateful to the A*STAR Research Attachment Programme (ARAP) of Singapore for financial support.

Author contributions

S.J.P., W.S.K., and H.K.L. contributed to the discussion and development of project. All authors approved the final manuscript.

Competing interest

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41598-020-70558-1.

Correspondence and requests for materials should be addressed to H.K.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.